2009-04-01

Acquisition Management for Systems-of-systems: Exploratory Model Development and Experimentation

Daniel DeLaurentis

NPS-AM-09-013

EXCERPT FROM THE
PROCEEDINGS

OF THE

SIXTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

ACQUISITION MANAGEMENT FOR SYSTEMS-OF-SYSTEMS: EXPLORATORY MODEL DEVELOPMENT AND EXPERIMENTATION

Published: 22 April 2009

by

Daniel DeLaurentis and Muharrem Mane

6th Annual Acquisition Research Symposium of the Naval Postgraduate School:

Volume I: Defense Acquisition in Transition

May 13-14, 2009

Prepared for: Naval Postgraduate School, Monterey, California 93943

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

The research presented at the symposium was supported by the Acquisition Chair of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request Defense Acquisition Research or to become a research sponsor, please contact:

NPS Acquisition Research Program
Attn: James B. Greene, RADM, USN, (Ret)
Acquisition Chair
Graduate School of Business and Public Policy
Naval Postgraduate School
555 Dyer Road, Room 332
Monterey, CA 93943-5103
Tel: (831) 656-2092
Fax: (831) 656-2253
E-mail: jbgreene@nps.edu

Copies of the Acquisition Sponsored Research Reports may be printed from our website www.acquisitionresearch.org

Conference Website:
www.researchsymposium.org

Proceedings of the Annual Acquisition Research Program

The following article is taken as an excerpt from the proceedings of the annual Acquisition Research Program. This annual event showcases the research projects funded through the Acquisition Research Program at the Graduate School of Business and Public Policy at the Naval Postgraduate School. Featuring keynote speakers, plenary panels, multiple panel sessions, a student research poster show and social events, the Annual Acquisition Research Symposium offers a candid environment where high-ranking Department of Defense (DoD) officials, industry officials, accomplished faculty and military students are encouraged to collaborate on finding applicable solutions to the challenges facing acquisition policies and processes within the DoD today. By jointly and publicly questioning the norms of industry and academia, the resulting research benefits from myriad perspectives and collaborations which can identify better solutions and practices in acquisition, contract, financial, logistics and program management.

For further information regarding the Acquisition Research Program, electronic copies of additional research, or to learn more about becoming a sponsor, please visit our program website at:

www.acquistionresearch.org

For further information on or to register for the next Acquisition Research Symposium during the third week of May, please visit our conference website at:

www.researchsymposium.org

THIS PAGE INTENTIONALLY LEFT BLANK

Acquisition Management for Systems-of-systems: Exploratory Model Development and Experimentation

Presenter: Daniel DeLaurentis is an Assistant Professor in the School of Aeronautics and Astronautics Engineering, Purdue University. He received his PhD from Georgia Institute of Technology in Aerospace Engineering in 1998. His current research interests are in Mathematical modeling and object-oriented frameworks for the design of system-of systems (especially those for which air vehicles are a main element), and approaches for robust design—including robust control analogies and uncertainty modeling/management in multidisciplinary design.

Daniel DeLaurentis
Assistant Professor, School of Aeronautics and Astronautics
Purdue University
701 W. Stadium Avenue, West Lafayette, IN 47907
Ph: 765-494-0694 Fax: 765-494-0307
Email: ddelaure@purdue.edu

Author: Muharrem Mane is a graduate student in the School of Civil Engineering, Purdue University. He received his PhD from Purdue University in Aerospace Engineering in 2008. He is currently pursuing a MS degree in Transportation Engineering and conducting research in the System-of-systems Research Group led by Dr. Daniel DeLaurentis.

Muharrem Mane
Graduate Research Assistant, School of Aeronautics and Astronautics
Purdue University
701 W. Stadium Avenue, West Lafayette, IN 47907
Ph: 765-494-7958 Fax: 765-494-0307
Email: mane@purdue.edu

Abstract

In recent years, the Department of Defense (DoD) has placed a growing emphasis on the pursuit of agile capabilities via net-enabled operations. In this setting, systems are increasingly required to interoperate along several dimensions. Yet, the manner in which components of these "system-of-systems" (SoS) are acquired (designed, developed, tested and fielded) has not kept pace with the shifts in operational doctrine. Acquisition programs have struggled with complexities in both program management and engineering design. We have developed a conceptual model for pre-acquisition and acquisition strategy in an SoS environment and have implemented it in an exploratory, dynamic model. The model allows acquisition professionals to develop intuition for procuring and deploying system-of-systems by providing a venue for experimentation through which they can develop insights that will underpin successful acquisition of SoS-oriented defense capabilities. This paper presents example studies that demonstrate the capabilities of the dynamic model and highlight the importance of project characteristics. Specifically, we investigate the impact of SoS attributes—requirement interdependency, project risk, and span-of-control of SoS managers and engineers—on the completion time of SoS projects.

Introduction

A system-of-systems (SoS) consists of multiple, heterogeneous, distributed systems that can (and do) operate independently but can also assemble in networks and collaborate to achieve a goal. According to Maier (1998), the SoS typically demonstrate traits of operational and managerial independence, emergent behavior, evolutionary development and geographic distribution. Networks of component systems often form among a hierarchy of levels and evolve over time as systems are added to or removed from the SoS. However, these component systems are often developed outside of the context of their interactions with the future SoS. As a result, the systems may be unable to fully interact with the future SoS, adapt to any emergent behavior, or be robust in the face of external disturbances.

The Future Combat System (FCS) program exemplifies a Department of Defense (DoD) acquisition process for an SoS. FCS seeks to modernize the US Army and provide soldiers with leading-edge technologies and capabilities—allowing them to dominate in asymmetric ground warfare and to sustain themselves in remote places (US Army, 2009). The FCS has faced technical and management challenges that have come to typify acquisitions in SoS environments.

In 2003, the FCS program was comprised of an information network and 18 primary systems (categorized as manned ground systems, unmanned ground systems, and unmanned air vehicles). The Army's initial schedule allotted a 56-month system development and demonstration (SDD) phase (2003-2008), with the goal of achieving full operational capability by 2013. The Army's initial cost estimate was $108 billion (GAO, 2003). Over the past four years, the FCS has been restructured twice in an effort to reduce the high risk attributed to both the presence of immature technologies in critical paths as well as the challenges of concurrently developing these technologies with product development. The Government Accountability Office (GAO) criticized the Army's acquisition strategy and concluded that the total cost for the FCS program had increased by 76% ($160.7 billion) from the Army's first estimate of $108 billion. However, independent estimates predicted an increase to $234 billion (116%).

In addition to the technical challenges, the FCS program also faced managerial challenges stemming from the Army's partnership with an industry Lead System Integrator (LSI). The role of the LSI is to reach across Army organizations to manage development of the SoS (GAO, 2007, June). Given the high risk involved in implementing a complex SoS, the GAO specifically underlined the importance of oversight challenges faced by the LSI in this area (GAO, 2007, March). The challenges of the FCS Program have pushed the Army to decrease the scope of the program to 14 systems and to extend the time estimate for achieving full capability to 2030 instead of 2013.

Other non-DoD organizations are also struggling with systems integration of a collection of complex systems. The US Coast Guard's (USCG) Integrated Deepwater System (IDS) is an example of a Department of Homeland Security (DHS) acquisition process for an SoS that has also faced challenges. These challenges have stemmed from the lack of collaboration between contractors and the marginal influence wielded by system integrators to compel decisions between them (GAO, 2006). The NextGen Air Transportation System and the NASA Constellation program are also facing similar challenges as they attempt to apply generic system engineering processes for acquisition in an SoS environment. Integration challenges faced by the Constellation Program are documented in a recent NRC report (Committee on System Integration for Project Constellation, 2004). These examples possess the key drivers motivating the research described in this paper.

The overarching goal of our research is to understand the types of complexities present in acquisition management for SoS, and then to develop approaches that can increase the success of an acquisition process in the SoS setting. The three research questions derived from this goal are:

1. Is there a taxonomy by which one can *detect* classes of complexities in particular SoS applications?
2. What are the underlying systems engineering (SE) and program management functions that are affected?
3. How can exploratory modeling generate SE and acquisition management modifications to improve the probability of success?

In order to answer some of the questions posed, we aim to:

1. Identify the complexities in the acquisition of SoS based on historical trends of "failures," especially in the context of the DoD
2. Develop a conceptual model of a generic acquisition process that is customizable to different SoS applications.
3. Develop a computational model based on the conceptual model and, through simulation, provide insight on and answer questions about process modifications.

Complexities

Simon (1996) and Bar-Yam (2003) define complexity as the amount of information necessary to describe a system effectively. In the context of a system-of-systems, the necessary information encompasses both the systems that comprise the SoS and their time-varying interactions with each other and the "externalities." Rouse (2007) summarized that the complexity of a system (or model of a system) is related to: the intentions with which one addresses the systems, the characteristics of the representation that appropriately accounts for the system's boundaries, architecture, interconnections and information flows, and the multiple representations of a system—all of which are simplifications. Hence, complexity is inevitably underestimated and context-dependent. Polzer, DeLaurentis, and Fry (2007) explored the issue of multiplicity of perspectives, in which perspective is a system's version of operational context.

Historical data from previous unsuccessful defense acquisition programs show a distinct correlation with the causes for complexity identified by Fowler (1994). Such data suggest some of the causes for the failure of the Defense Acquisition Process to be "over specification and an overly rigid approach on development," unreasonably detailed cost estimates of development and production, impractical schedules, and extremely large bureaucratic overhead. Dr. Pedro Rustan, Director of Advanced Systems and Technology at the National Reconnaissance Office, identified four specific shortcomings in the acquisition process for defense space systems: initial weapons performance requirements that are too detailed and lacking flexibility, insufficient flexibility in the budget process, a propensity to increase performance requirements in the middle of the acquisition cycle, and demands to field entirely new spacecraft to meet new requirement (Spring, 2005).

Using the above examples, we summarize the common causes of failure (Rouse, 2007) within SoS acquisition processes as: a) *misalignment* of objectives among the systems, b) limited *span-of-control* of the SoS engineer on the component systems of the SoS, c) *evolution* of the SoS, d) *inflexibility* of the component system designs, e) *emergent behavior* revealing hidden dependencies within systems, f) *perceived complexity* of systems and g) the challenges in *system representation*.

To provide context, in Ghose and DeLaurentis (2008), we mapped these complexities to a System-of-systems Engineering (SoSE) Process Model designed specifically for SoS applications by Sage and Biemer (2007). This mapping represents points at which complexities might arise and how they may affect the acquisition process.

Development of a Conceptual Model

Pre-acquisition Model

We developed a pre-acquisition model to understand the impact of external stakeholders on the acquisition process. The model is based loosely on the Sage and Biemer (2007) SoSE Process Model and categorizes the external inputs to the SoS acquisition strategy model into "Capabilities & Possibilities" (CAP), "Technology Assessment, Development, Investment and Affordability Plan" (ADIA) and the funding received (Ghose & DeLaurentis, 2008). The CAP and the Technology ADIA Plan translate into technical requirements for the SoS. Provision of a computational model of the pre-acquisition activities is outside the scope of this paper. Instead, we focus on realizing a model for the acquisition strategy, described next.

Acquisition Strategy Model

Development of a "brand new" SoS has been and will remain a rare occurrence. In their 2005 study on SoS, the United States Air Force (USAF) Scientific Advisory Board (Saunders et al., 2005) stated that one of the challenges in building an SoS is accounting for contributions and constraints of legacy systems. These legacy systems may be used "as-is" or may need re-engineering to fulfill the needs of the new SoS. New systems are also incorporated to develop the capabilities of the SoS. Again, the new systems may range from off-the-shelf, plug-and-play products to custom-built systems dependent of the working of a legacy system. Sub-categories arise when the two or more categories overlap (Figure 2).

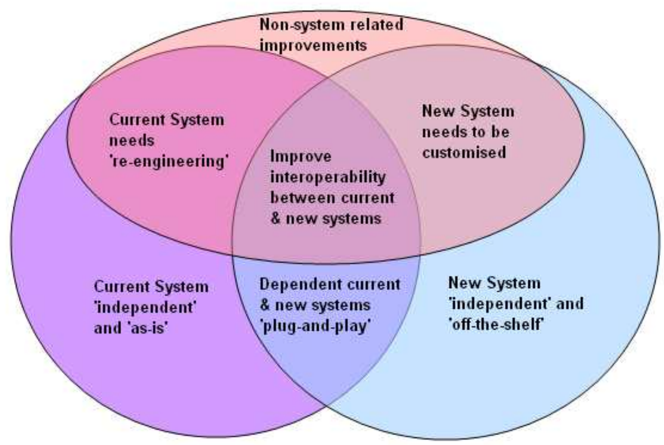


Figure 1. Heterogeneity of Component Systems in an SoS

The conceptual model for acquisition strategy proposed in this section is based on the 16 basic technical management and technical system-engineering processes outlined in the *Defense Acquisition Guidebook* (DoD, 2003), often referred to as the 5000-series guide. However, an SoS environment changes the way these processes are applied. The *Systems Engineering Guide for System-of-Systems* (*SoS-SE*) (DoD, 2008) addresses these considerations by modifying (in some cases revamping) some of the 16 processes in accord with an SoS environment. These new processes and their functions are described in Table 1. Our conceptual model for acquisition in an SoS environment (illustrated in Figure 3) is centered on these revised processes, depicted in a hierarchy to show the flow of control between the processes throughout the acquisition lifecycle.

Table 1. Modified Technical Management and Technical Processes as Described in the *SoS-SE Guide*
(DoD, 2003)

Requirements Development	Takes all inputs from relevant stakeholders and translates the inputs into technical requirements.
Logical Analysis	Obtains sets of logical solutions to improve the understanding of the defined requirements and the relationships among the requirements (e.g., functional, behavioral, temporal).
Design Solution	Translates the outputs of the Requirements Development and Logical Analysis processes into alternative design solutions and selects a final design solution.
Decision Analysis	Provides the basis for evaluating and selecting alternatives when decisions need to be made.
Implementation	Yields the lowest-level system elements in the system hierarchy. The system element is made, bought or reused.
Integration	Incorporates the lower-level system elements into a high-level system element in the physical architecture.
Verification	Confirms that the system element meets the design-to or build-to specifications. It answers the question "Did you build it right?"
Validation	Answers the question of "Did you build the right thing?"
Transition	Applies the process required to move the end-item system to the user.
Technical Planning	Ensures that the systems engineering processes are applied properly throughout a system's lifecycle.
Technical Assessment	Measures technical progress and the effectiveness of plans and requirements.
Requirements Management	Provides traceability back to user-defined capabilities
Risk Management	Helps ensure program cost, schedule and performance objectives are achieved at every stage in the lifecycle and communicates to all stakeholders the process for uncovering, determining the scope of, and managing program uncertainties.
Configuration Management	Ensures the application of sound business practices to establish and maintain consistency of a product's attributes with its requirements and product configuration information.
Data Management	Addresses the handling of information necessary for or associated with product development and sustainment.
Interface Management	Ensures interface definition and compliance among the elements that compose the system, as well as with other systems with which the system or systems elements must interoperate.

A detailed description of the conceptual model and the acquisition stages it models (Figure 2) is presented in Ghose and DeLaurentis (2008).

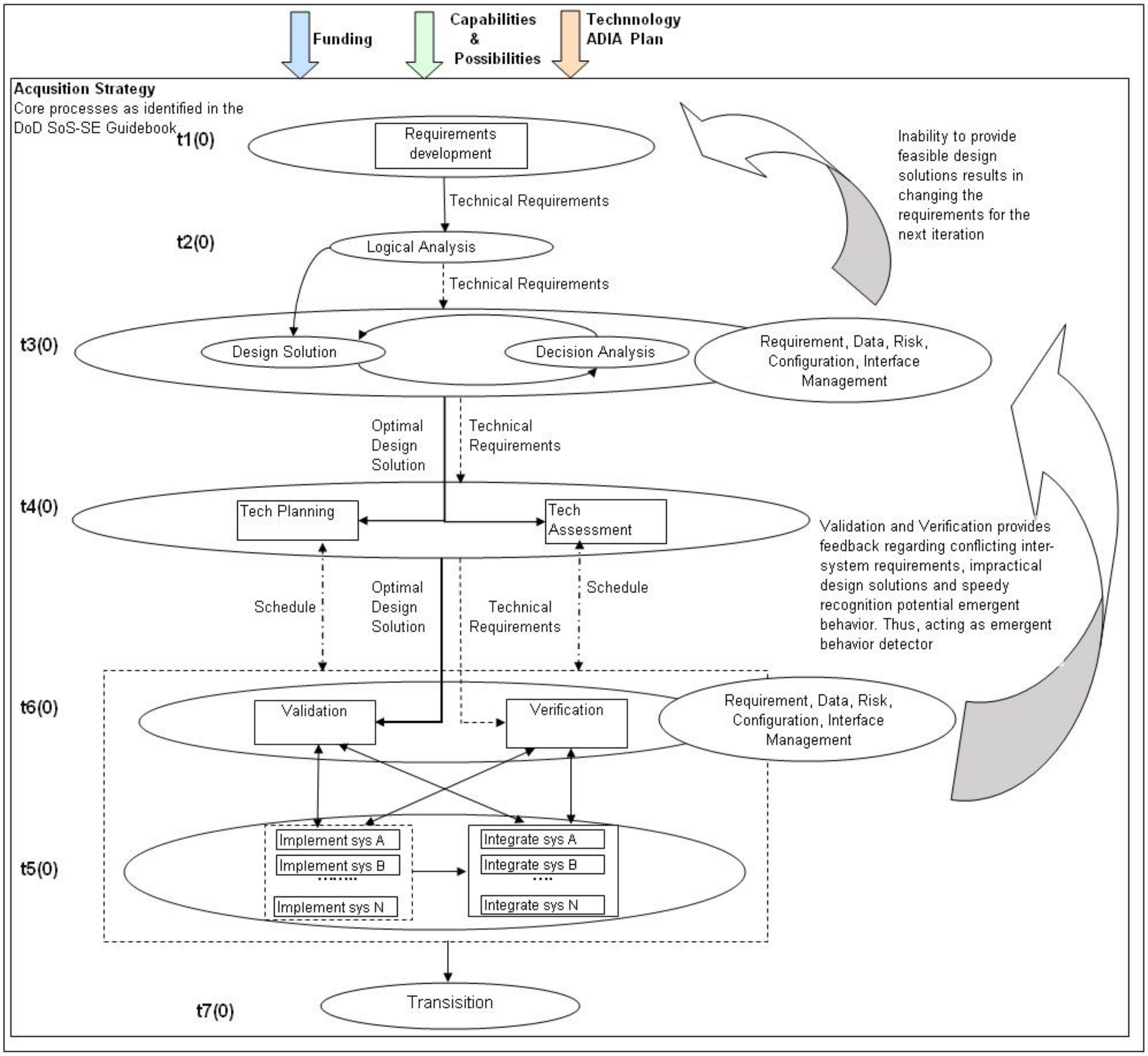


Figure 2. Conceptual Model of Acquisition Strategy Based on SoSE Process Described in Table 1

The purpose of the exploratory computational model is to help acquisition professionals develop intuition for procuring and deploying systems in a system-of-systems context, not to provide a tool validated for use in managing real acquisition programs. A model that captures all the complexity of the acquisition process for SoS in a modest span of time and effort is impossible. The exercise of the model described in this paper specifically targets complexities stemming from the interdependencies among systems, the evolutionary development of the SoS and the span-of-control of the SoS managers and engineers. An abstraction of the model is presented in Figure 3.

At the requirement level, each node represents a requirement, while each link represents the interdependency between requirements. Similarly, at the system level, each node represents a system and each link the interdependency between systems. Groupings of interdependent systems are needed to fulfill a requirement. In our computational model, the user can specify the number of requirements and their interdependencies as well as the number of systems and their interdependencies for each requirement, or the user can randomly generate the requirement and system interdependencies. It is with this layered network concept/representation that the computational model progresses through the acquisition stages described in Figure 2.

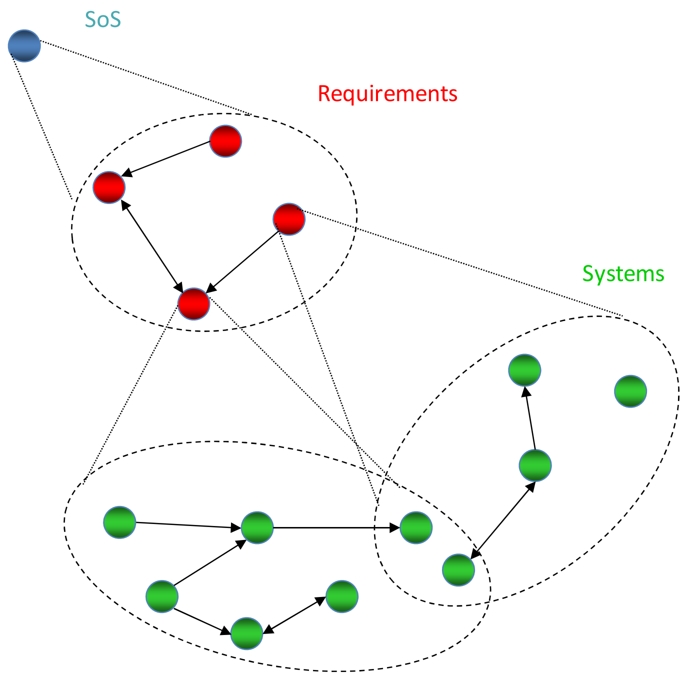


Figure 3. Node/link Picture of Acquisition Model

Developing the Exploratory Computational Model

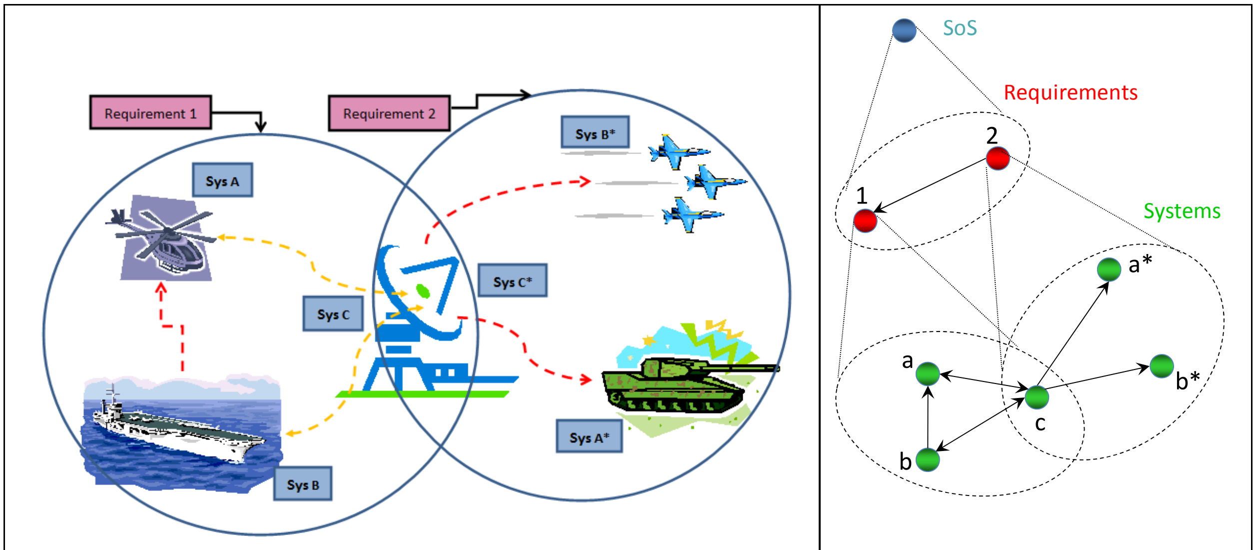
Overview

Several challenges arise in transforming the acquisition model to a computational one for the purposes of simulation and learning. One challenge lies in converting all the qualitative concepts into quantitative measures to support the computational model for SoS acquisition. Disruptions occur at various stages in the model and are governed by the risk associated with the project. A high-risk project, for example, will be more vulnerable to disruptions than a low-risk project. A second challenge is building a model that can accommodate the dynamic addition and removal of components in the SoS. In addition, these component systems need to reflect the heterogeneity of the systems in a real acquisition process. We included parameters such as *level of completeness* to demonstrate the difference between legacy systems, new systems and partially implemented/integrated systems. A third challenge arises from the numerous methodologies that can be applied to reflect the integration and implementation processes. In a simplified model, it is much easier to begin integration once all the systems have been implemented. However, this method is neither cost- nor time-efficient, especially in multi-year projects involving numerous systems. On the other hand, dynamically implementing and integrating systems is time-efficient but often not possible when dependent systems are outside the span-of-control of the systems' engineers.

As stated previously, a model that captures all the complexity of the acquisition process for SoS in a modest span of time is impossible. Therefore, our coarse-scale engineering model will specifically target challenges related to the evolution of the SoS and the span-of-control of the SoS engineer(s).

Simple SoS Example

A simple SoS acquisition strategy with two requirements and five component systems (Figure 4) is first presented to illustrate the model workings. Figure 4(a) shows the physical composition of the SoS, while Figure 4(b) presents the layered network of this simple SoS.



a) Example of SoS b) Model Structure for Example SoS

Figure 4. Simple Example of SoS

Requirement 1 is to improve rescue operations performed by a certain fleet, while Requirement 2 is to improve communication and coordination between air and ground units. The three types of component systems fulfilling Requirement 1 are helicopter (A), ship (B) and communication system (C). Similarly, the three component systems fulfilling Requirement 2 are ground units (A*), airborne units (B*) and a communication system (C*).

Since Requirement 2 needs to use the communication system (C) built by Requirement 1, Requirement 2 is dependent on Requirement 1. The directional dependencies within the component systems fulfilling each requirement are shown in Figure 4(a) using dashed yellow (bidirectional) and red (unidirectional) lines. The requirement level dependency matrix and the system-level dependency matrices for each requirement are shown in Table 2.

Table 2. Dependency Matrices

Requirement Dependency Matrix	$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

Requirement	System Dependency Matrix
1	$\begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$
2	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$

Model Inputs

Three levels of inputs are used in the model: project-level, requirement-level and system-level. The three user-defined project-level inputs are project-risk, span-of-control of SoS managers and engineers, and estimated amount of time needed to complete the project. A project can have low, medium or high project-risk profile. This profile determines: a) the probability of the project being affected by disruptions at *Design Solution* (Level t3(0), Figure 2) and *Implementation & Integration* (Level t5(0), Figure 2) stage, and b) the probability of a new requirement being added during the project lifecycle. The span-of-control of an SoS engineer or manager indicates whether component systems are directly or indirectly accountable to the SoS manager or engineer. A project's span-of-control is either "0" or "1," where "0" represents low span-of-control. A project with low span-of-control implements dependent systems sequentially instead of in parallel. The requirement-level inputs to the exploratory computational model are initial number of requirements, dependencies between requirements, component systems fulfilling each requirement, and the dependencies between the component systems.

The dependencies between the requirements determine the schedule by which the requirements will be implemented. For the simple example problem, as shown in Table 2, there are two requirements (1, 2), and each has a dependency vector associated with it. The vectors are concatenated to form the dependency matrix for requirements ("0" is placed for all diagonal elements since a requirement cannot be dependent on itself). The vector for Requirement 1 ([0 1]) shows that Requirement "1" is dependent on Requirement "2," and "1" cannot be realized until "2" is implemented. In real-world applications, communication upgrade to the North-Atlantic fleet may be independent of the weaponry upgrade for the same group of systems. In such a case, both the requirements on the same group of systems may be implemented simultaneously. Each requirement affects a subset of the systems present in the SoS, and the systems in each subset share a unique dependency matrix with other systems in that subset.

All component systems of the SoS have user-defined and calculated system-level parameters that expose their heterogeneity and help track their progress through the acquisition process. Some of the parameters used to describe each system in the SoS are described in Table 3.

Table 3. System-level Parameters Used to Describe Component System of the SoS

Parameter	Description
ID	Unique ID assigned to the system
Imp.completeness[]	An array that tracks the progress of the system in the implementation phase
Imp.dependencies[]	Dependency vector that shows if system implementation is dependent on information from any other system
Imp.time	Maximum time needed to complete implementation
Int.completeness[]	An array that tracks the progress of the system in the integration phase
Int.dependencies[]	Dependency vector that shows if system integration is dependent on information from any other system
Int.time	Maximum time needed to complete integration

While most of the parameters are user-defined, Imp.completeness and Int.completeness are only initialized by the user, and ID is assigned by the model. Implementation or Integration of a system[A] is either dependent on information from other systems satisfying the requirement or independent of any such information. Thus, all the tasks necessary to successfully implement or integrate system[A] can be divided into smaller subsets depending upon which systems they need information from. At a given time-step, the *level of completeness* of system[A] with regard to system[X] is defined as the percent of tasks needed to successfully implement/integrate system[A] that are dependent on information from system[X] and have been completed. *Level of completeness* for both integration and implementation processes can vary between 0 and 100%. The *level of completeness* of system[A] with regard to *N* individual systems is summed to calculate the total *level of completeness* of system[A]. Note that although the tasks are dependent on information from system[A], the *level of completeness* says nothing about the status of system[A]. Note also that the model works in discrete time.

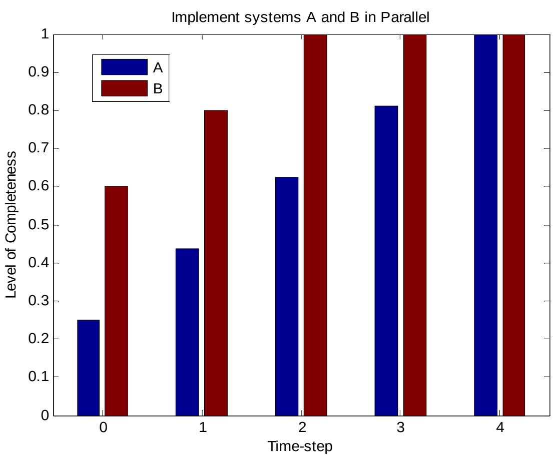
Similar to requirements, each system has a pre-defined dependency vector for implementation and integration processes. These vectors are concatenated to form a dependency matrix for the systems fulfilling each requirement.

Model Dynamics

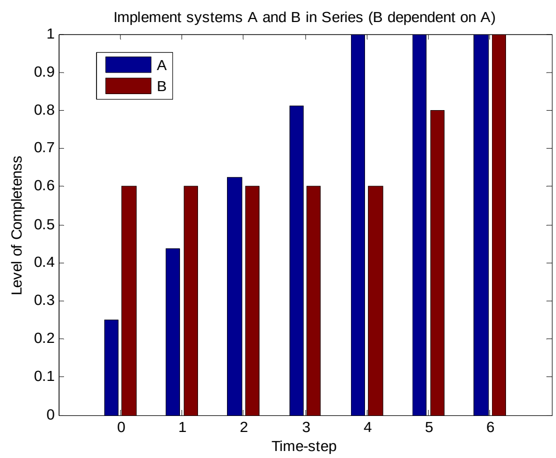
The model begins at the *Requirement Development* (Level t0(0), Figure 3) stage, which initializes requirements to be implemented, project span-of-control and project risk. Disruptors at the requirement level can take the form of change in existing requirements or addition of new requirements. The user-defined inputs from *Requirement Development* are passed to *Logical Analysis* (Level t2(0), Figure 2), which generates a schedule to realize the given requirements either in series or in parallel (per the dependencies). Each requirement then enters its own *Design Solution* and *Decision Analysis* (Level t3(0), Figure 2) process. The *Design Solution* and *Decision Analysis* processes feed into each other, and any disruptions at this stage imply that the design solution provided is not feasible. If the solution fails in multiple consecutive time-steps, then the requirement is sent back to the *Requirement Development* stage; otherwise, the set of component systems and their user-defined parameters are sent to the *Technology Planning* and *Technology Assessment* (Level t4(0), Figure 2) processes.

Implementation (Level t5(0), Figure 2) of systems occur in series or parallel, depending on the system dependencies and the span-of-control of the project. The *level of completeness* for implementation increases by the iteration rate at every time-step until it reaches a completeness value of 1. The incremental increase in the level of completeness of two dependent systems in a project with high span-of-control ("1") occurs simultaneously, as shown in Figure 5(a). In a case of low span-of-control ("0"), dependent systems are implemented sequentially, as shown in Figure 5(b).

When a system achieves the implementation completeness = 1, it enters the *Integration* stage.



a) Independent

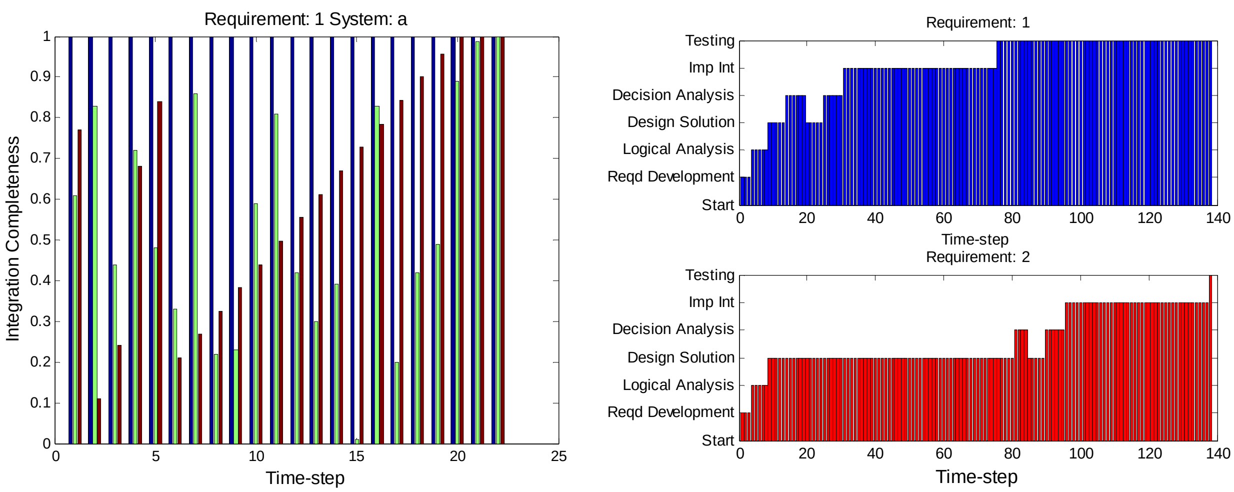


b) Dependent Systems

Figure 5. Incremental Increase in Implementation Completeness

Similar to *Implementation*, systems can be integrated in series or in parallel depending on the span-of-control. When both the *Implementation* and *Integration* processes for the given requirement are complete, the *Validation* and *Verification* phase (Level t6(0), Figure 2) checks for a completeness level of “1” for all component systems. If the requirement successfully passes *Validation* and *Verification*, it is said to be ready for *Testing*. A more detailed description of these stages is presented by Ghose and DeLaurentis (2008).

To present an example of output generated by the computational model, we simulate the acquisition process of the simple SoS presented in Figure 4. We assume that this project has a high span-of-control and a low risk level. All systems have random initial completeness levels as well as implementation and integration times. Results for this simple example from the computational model are presented in Figure 6. Results similar to the ones presented on the left plot are available for all systems that comprise the acquisition project in this example.



a) Integration completeness of system “a”

b) Completeness history of entire project

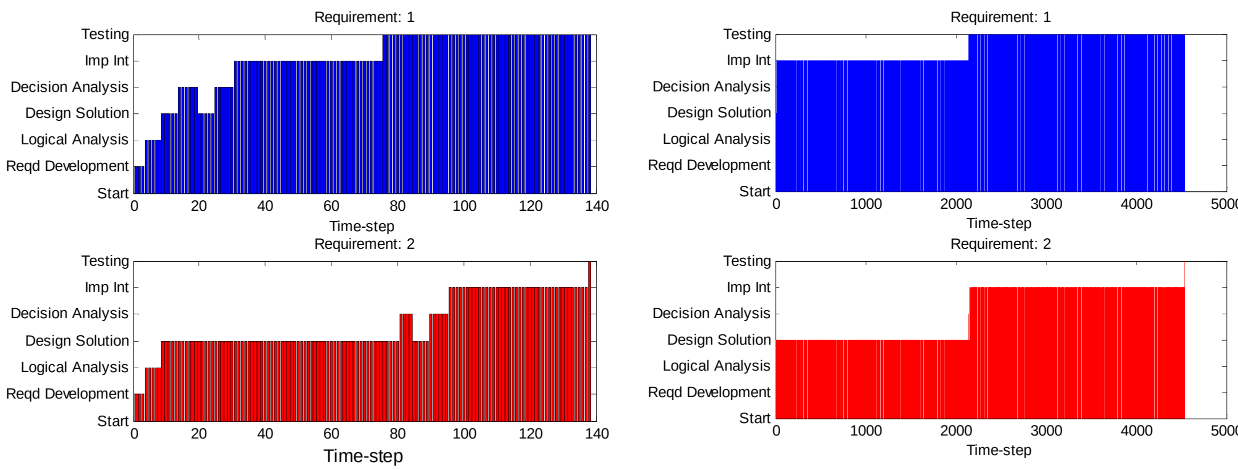
Figure 6. Sample Results of Computational Model for Example Problem

In Figure 6(a), each bar represents a system that is part of requirement 1. Because we are observing system “a,” its integration with itself has a value of “1.” The integration completeness of system “a” with systems “b” and “c” fluctuates (due to disruptions—occurring here with a uniformly random probability) until after 22 time-steps, at which point integration is complete. The numerous set-backs in integrating systems “b” and “c” indicate key dynamic features of this model. Though modeled as uniformly random here, we envision more meaningful probability functions for the occurrence of disruptions that relate to physical or actual observed patterns. When the system histories are compiled, the result is the acquisition process history shown in Figure 6(b). Evidence of the impact of disruptions on completeness is noticeable. The completion time of this acquisition project is 138 time units. Note, however, that requirement 2 shows no activity after the *Design Solutions* phase from 10 to 81 time units; requirement 2 is dependent on requirement 1, which is completed after 81 time units.

Case Studies

Management organization and the complexity of requirements vary from SoS project to project. Further, component systems that comprise the SoS have different risk levels that add to the complexity and uncertainty of a given SoS. In these case studies, we utilize the exploratory model to test the dynamics underlining the acquisition management in an SoS environment. We explore the impact of span-of-control, requirement dependency, and system risk on the completion-time of an SoS. First, we study the impact of span-of-control by simulating the acquisition process for low and high span-of-control. Then, we simulate twelve scenarios—which result from the combination of low and high span-of-control, dependent and independent requirements, and low, medium, and high risk profile—and study the impact of these project and system characteristics on the project's completion time.

The effect of span-of-control is studied by simulating the acquisition process of the example problem described in Figure 4 for low and high span-of-control. All the values of the input parameters are the same (same probability of occurrence of disruptions and low risk level) for each scenario, while the span-of-control is varied from low to high. Figure 7 present the results for these two scenarios.



a) High Span-of-control Result

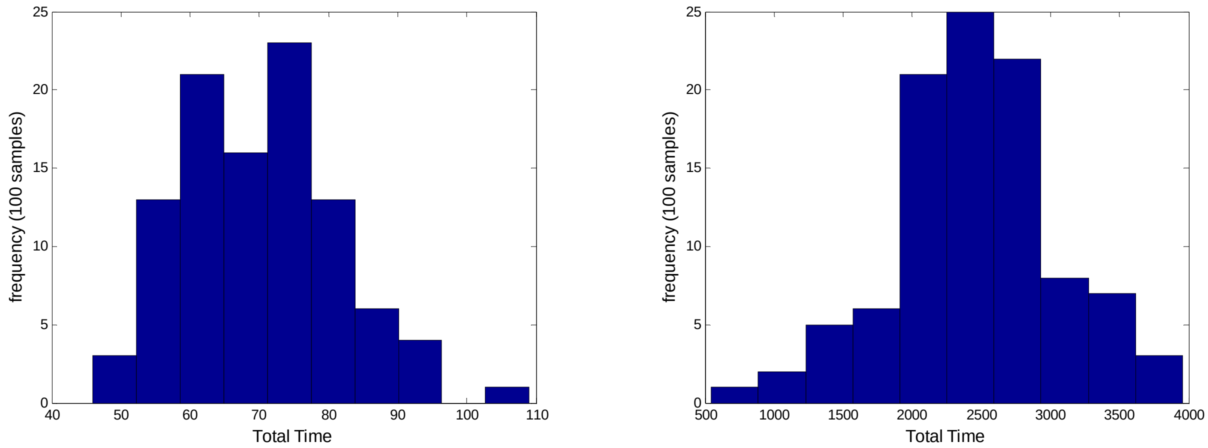
b) Low Span-of-control Result

Figure 7. Impact of Span-of-control

Because the example presented in Figure 6 already considered the high span-of-control scenario, the same result is presented here in Figure 7(a). Figure 7(b), on the other hand, presents the results of the scenario when the SoS has low span-of-control. The comparison of these two scenarios makes obvious the impact of the span-of-control parameter. For low span-of-control, the project completion time is about 4500 time units, while high span-of-control permits the completion of the same project in 138 time units.

Since the probability of disruptions is never zero, disruptions inevitably occur that impact the system completeness level and, ultimately, the project completion time. Because the model is probabilistic in nature, 100 different runs are performed for each scenario, and the mean completion time is recorded. To isolate the effect of the random disruptions, we enforce all systems to have the same initial completeness level for all 100 runs; furthermore, we assume that when a disruption occurs, it will not reduce the completeness level below the initial value.

Figure 8 presents a distribution of the completion time for each of these scenarios. As expected, the mean completion time when span-of-control is high (70 time units) is lower than when span-of-control is low (2,474 time units, a 35-fold increase).



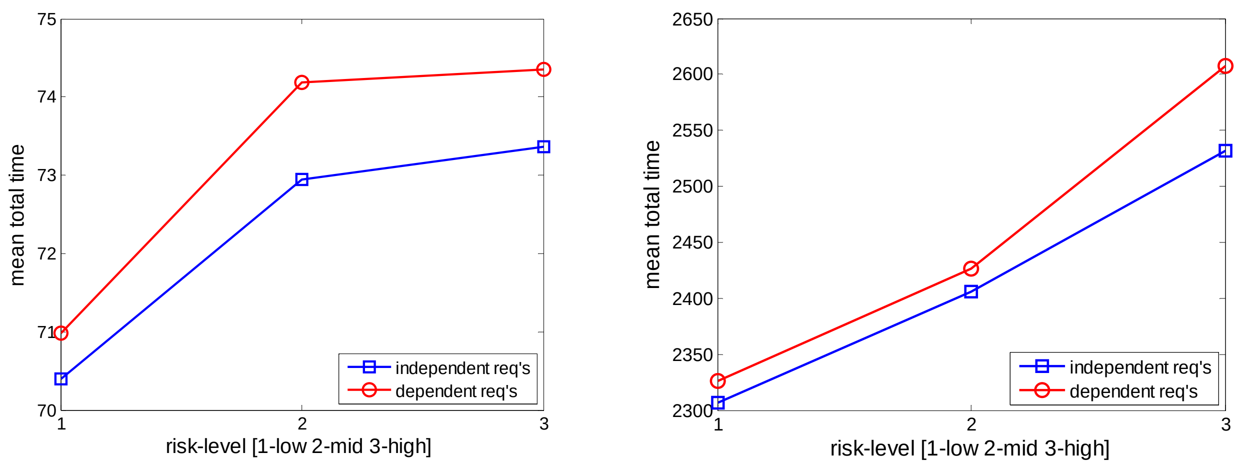
a) High Span-of-control Result

b) Low Span-of-control Result

Figure 8. Distribution for Completion Time for Low and High Span-of-control

This behavior seems reasonable when we consider that when the span-of-control is low, systems are integrated and implemented sequentially, which increases the probability of disruptions. The variance is also lower in the high span-of-control case.

As previously mentioned, the acquisition model also uses risk level to describe the probability of disruptions during the design of component systems. Its impact on the completion time when coupled with span-of-control and requirement interdependency is, thus, also investigated. Figure 9 displays the results for combinations of low and high span-of-control with low, medium, and high risk levels—all for cases of both dependent and independent requirements.



a) High Span-of-control Results

b) Low Span-of-control Results

Figure 9. Comparison of Project and System Characteristics

Each data point in Figure 9 represents the mean completion time after 100 runs. As expected, these mean total time results show that span-of-control has the largest impact on completion time. Additionally, the impact of dependent requirements is much greater in the low span-of-control case. A dependent requirement must wait for the completion of the requirement on which it depends, and when both requirements must sequentially implement and integrate component systems (low span-of-control), the result is a substantial increase in completion time.

The results from these twelve test cases are used next in a sensitivity analysis to quantify the relative importance of each of the three parameters on the total time needed to complete the project.

Sensitivity Analysis

Sensitivity analysis further investigates the impact of the three parameters (requirement dependency, span-of-control, and risk profile) studied in the twelve test cases.

Requirement Dependency: Compare completion time in cases of dependent versus independent requirements while keeping span-of-control and risk profile constant. Table 5 presents the ratio of the mean completion time of the scenarios with dependent requirements to the mean completion time of the scenarios with independent requirements. Risk profiles are labeled “1” for Low, “2” for Medium and “3” for High. These results show that scenarios with dependent requirements take marginally longer when compared to projects with independent requirements. Note, however, that as Figure 9 showed, for low span-of-control, the absolute increase in the mean completion time is still relatively large.

Table 5. Effect of Requirement Dependency

Span-of-control	Risk	Ratio	Span-of-control	Risk	Ratio
1	1	1.008	0	1	1.008
1	2	1.017	0	2	1.008
1	3	1.013	0	3	1.030

Span-of-Control: Compared cases of low versus high span-of-control while keeping requirement-dependency and risk profile constant. Table 6 presents the ratio of the mean completion time of the scenarios with low span-of-control to the mean completion time of the scenarios with high span-of-control. The six results indicate the level of risk of each scenario (labeled “1” for Low, “2” for Medium and “3” for High) and whether requirements are dependent or independent (labeled “I” for independent and “D” for dependent). These results show that low span-of-control increases the mean completion time by a factor of 32.70 to 35.08. Also of note is that the largest increases in completion time occur when requirements are dependent. This is an expected result because dependent requirements are completed sequentially instead of in parallel.

Table 6. Effect of Span-of-control

I/D	Risk	Ratio	I/D	Risk	Ratio
I	1	32.77	D	1	32.77
I	2	32.98	D	2	32.70
I	3	34.51	D	3	35.08

Risk Profile: Compared cases of three risk profiles, while keeping requirement dependency and span-of-control constant. Table 7 presents the ratio in mean completion time between scenarios with risk “2” and “3” and risk “1.” These ratios indicate that as risk increases, so does the mean completion time. As expected, the highest increase is observed for high risk levels (risk with value “3”) for both low and high span-of-control scenarios. For example: for a project with independent requirements and high span-of-control, the ratio of the mean completion time for a high risk (“3”) profile versus a low risk (“1”) profile is 1.042.

Table 7. Effect of Increasing Project Risk

I/D	Span-of-control	Risk	Ratio	I/D	Span-of-control	Risk	Ratio
I	1	1	-	I	0	1	-
I	1	2	1.036	I	0	2	1.043
I	1	3	1.042	I	0	3	1.098
D	1	1	-	D	0	1	-
D	1	2	1.045	D	0	2	1.043
D	1	3	1.047	D	0	3	1.121

Results

Some insights gained from testing the exploratory model via the sensitivity analysis are:

1. As expected, time to implement dependent requirements is always greater than the independent case; completion time strongly depends on the span-of-control of the SoS managers and engineers, as well as on the project risk.
2. Time needed to implement projects with higher risk profiles is always greater than the time needed to implement the project with lower risk profiles.
3. The sensitivity analysis shows that the time needed to complete a project is much more sensitive to the span-of-control of the SoS engineers and managers than to the project risk or the dependencies between the requirements.
4. A project with high span-of-control is better equipped to recover from the debilitating disruptions associated with a high risk, thus making the acquisition process more resilient.

Conclusions

We have developed a conceptual model for pre-acquisition and acquisition strategy activities by mapping the sources of complexity to a section of the SoSE Process Model by Sage and Biemer (2007) in conjunction with the 16 technical and technical-management SE processes identified by the *SoS-SE Guide* (DoD, 2008). This mapping and conceptual model provide a basis for a computational exploratory model for acquisition strategy in an SoS environment. The purpose of the model is to explore the complexities that arise in SoS acquisition programs due to evolutionary development of the SoS, heterogeneity of the component systems, as well as the effect of management parameters on the acquisition programs. Based on user-defined inputs for the requirements and their interdependencies, the model uses series and parallel processing to implement and integrate the component systems that comprise the SoS while allowing the impact of disruptors to propagate through the various processes in the acquisition hierarchy.

In this study, we use the dynamic exploratory model to investigate the impact of requirement interdependency, project risk, and span-of-control on the completion time of SoS projects. Results from test scenarios and sensitivity analysis underline the importance of span-of-control of SoS managers and engineers on the timely completion of projects. Projects with a low span-of-control always require more time to complete than projects with high span-of-control. Furthermore, the effects of requirement interdependency and project risk are always overshadowed by the impact of span-of-control. A high span-of-control positively affects completion time by making the acquisition process more resilient and agile in the face of disruptions. While some of these observations confirm intuition, the computational model provides a means to test acquisition and/or management strategies and explore new approaches for the SoS acquisition process.

The uniqueness of the models (both conceptual and computational) lies in their ability to provide decision-makers with a better understanding of the acquisition process in an SoS environment. The models also offer computational tools to aid decision-making for the higher levels of SoS management. We hope that the insights gained from this research will improve the probability of success of future acquisition programs of complex SoS.

Acknowledgements

The authors acknowledge the support for this work from a grant from the Naval Postgraduate School's Acquisition Research Program (N00244-09-1-0011). We also recognize the contributions of Shayani Ghose, presently with Rockwell Collins in Cedar Rapids, IA, for her contributions toward development of the computational model.

List of References

Bar-Yam, Y. (2003). *Complexity of military conflict: Multiscale complex systems analysis of littoral warfare.* New England Complex Systems Institute. Retrieved from http://www.necsi.edu/projects/yaneer/SSG_NECSI_3_Litt.pdf

Committee on Systems Integration for Project Constellation. (2004). *Systems integration for project constellation*. The National Academies. Retrieved from http://nap.edu/html/proj-constellation/ltr-rep.pdf

DoD. (2003, May 12). *Defense acquisition guidebook.* Online, GPO Access. Retrieved April 4, 2008, from https://akss.dau.mil/dag/

DoD. (2008, January). *Systems engineering guide for system-of-systems*. Online, GPO Access. Retrieved March 2, 2008, from http://www.acq.osd.mil/sse/ssa/initiat_sos-se.html\

Fowler, C.A. (1994). The defense acquisition system: Too late for the scalpel; Bring out the meataxe! *IEEE Aerospace and Electronic Systems Magazine, 9*(8), 3-6.

GAO. (2003, August 13). *Issues facing the Army's Future Combat Systems.* Online, GPO Access. Retrieved rom http://www.gao.gov/new.items/d031010r.pdf

GAO. (2006, June). *Coast Guard: Observations on agency performance, operations and future challenges*. Retrieved from http://www.gao.gov/new.items/d06448t.pdf

GAO. (2007, March 27). *Future Combat System risks underscore the importance of oversight*. Online, GPO Access. Retrieved from http://www.gao.gov/new.items/d07672t.pdf

GAO. (2007, June). *Role of lead systems integrator on Future Combat Systems program poses oversight challenge.* Online, GPO Access. Retrieved from http://www.gao.gov/new.items/d07380.pdf

Ghose, S., & DeLaurentis, D.A. (2008, May 14-15). Defense acquisition management of systems-of-systems. In *Proceedings of the 5th Annual Acquisition Research Symposium.* Monterey, CA: Naval Postgraduate School.

Ghose, S., & DeLaurentis, D.A. (2008, June 15-19). Development of an exploratory model for acquisition management of system-of-systems. 18th Annual International Symposium of INCOSE, Ultrecht, The Netherlands.

Maier, M. (1998). Architecting principles for system-of-systems. *Systems Engineering, 1*(4), 267-284.

Polzer, H., DeLaurentis, D.A., & Fry, D.N. (2007). Multiplicity of perspectives, context scope, and context shifting events. In *Proceedings of the 2007 IEEE International Conference on System of Systems Engineering* (pp. 1-6).

Rouse, W. (2007). Complex engineered, organizational and natural systems. *Systems Engineering, 10*(3), 260-271.

Sage, A., & Biemer, S. (2007). Processes for system family architecting, design, and integration. *IEEE Systems Journal, 1*(1), 5-16.

Saunders, T., Croom, C., Austin, W., Brock, J., Crawford, N., Endsley, M., et al. (2005). *Report on system-of-systems engineering for Air Force capability development* (SAB-TR-05-04). Washington, DC: USAF Scientific Advisory Board.

Simon, H. (1996). *Sciences of the artificial*. Cambridge, MA: MIT Press.

Spring, B. (2005). Congressional restraint is key to successful defense acquisition reform. *The Heritage Foundation.* Retrieved March 26, 2008, from http://www.heritage.org/Research/NationalSecurity/bg1885.cfm

US Army. (2009). *Future combat systems.* Retrieved from https://www.fcs.army.mil/

THIS PAGE INTENTIONALLY LEFT BLANK

2003 - 2009 Sponsored Research Topics

Acquisition Management

- Acquiring Combat Capability via Public-Private Partnerships (PPPs)
- BCA: Contractor vs. Organic Growth
- Defense Industry Consolidation
- EU-US Defense Industrial Relationships
- Knowledge Value Added (KVA) + Real Options (RO) Applied to Shipyard Planning Processes
- Managing Services Supply Chain
- MOSA Contracting Implications
- Portfolio Optimization via KVA + RO
- Private Military Sector
- Software Requirements for OA
- Spiral Development
- Strategy for Defense Acquisition Research
- The Software, Hardware Asset Reuse Enterprise (SHARE) repository

Contract Management

- Commodity Sourcing Strategies
- Contracting Government Procurement Functions
- Contractors in 21st Century Combat Zone
- Joint Contingency Contracting
- Model for Optimizing Contingency Contracting Planning and Execution
- Navy Contract Writing Guide
- Past Performance in Source Selection
- Strategic Contingency Contracting
- Transforming DoD Contract Closeout
- USAF Energy Savings Performance Contracts
- USAF IT Commodity Council
- USMC Contingency Contracting

Financial Management

- Acquisitions via leasing: MPS case
- Budget Scoring
- Budgeting for Capabilities-based Planning
- Capital Budgeting for DoD

- Energy Saving Contracts/DoD Mobile Assets
- Financing DoD Budget via PPPs
- Lessons from Private Sector Capital Budgeting for DoD Acquisition Budgeting Reform
- PPPs and Government Financing
- ROI of Information Warfare Systems
- Special Termination Liability in MDAPs
- Strategic Sourcing
- Transaction Cost Economics (TCE) to Improve Cost Estimates

Human Resources

- Indefinite Reenlistment
- Individual Augmentation
- Learning Management Systems
- Moral Conduct Waivers and First-tem Attrition
- Retention
- The Navy's Selective Reenlistment Bonus (SRB) Management System
- Tuition Assistance

Logistics Management

- Analysis of LAV Depot Maintenance
- Army LOG MOD
- ASDS Product Support Analysis
- Cold-chain Logistics
- Contractors Supporting Military Operations
- Diffusion/Variability on Vendor Performance Evaluation
- Evolutionary Acquisition
- Lean Six Sigma to Reduce Costs and Improve Readiness
- Naval Aviation Maintenance and Process Improvement (2)
- Optimizing CIWS Lifecycle Support (LCS)
- Outsourcing the Pearl Harbor MK-48 Intermediate Maintenance Activity
- Pallet Management System
- PBL (4)
- Privatization-NOSL/NAWCI
- RFID (6)
- Risk Analysis for Performance-based Logistics
- R-TOC Aegis Microwave Power Tubes

- Sense-and-Respond Logistics Network
- Strategic Sourcing

Program Management

- Building Collaborative Capacity
- Business Process Reengineering (BPR) for LCS Mission Module Acquisition
- Collaborative IT Tools Leveraging Competence
- Contractor vs. Organic Support
- Knowledge, Responsibilities and Decision Rights in MDAPs
- KVA Applied to Aegis and SSDS
- Managing the Service Supply Chain
- Measuring Uncertainty in Earned Value
- Organizational Modeling and Simulation
- Public-Private Partnership
- Terminating Your Own Program
- Utilizing Collaborative and Three-dimensional Imaging Technology

A complete listing and electronic copies of published research are available on our website: www.acquisitionresearch.org

THIS PAGE INTENTIONALLY LEFT BLANK

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CALIFORNIA 93943

www.acquisitionresearch.org